2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

A Heuristic for Free Parameter Optimization with Support Vector Machines

Matthew Boardman and Thomas Trappenberg

***Abstract*— A heuristic is proposed to address free parameter selection for Support Vector Machines, with the goals of improving generalization performance and providing greater insensitivity to training set selection. The many local extrema in these optimization problems make gradient descent algorithms impractical. The main point of the proposed heuristic is the inclusion of a model complexity measure to improve generalization performance. We also use simulated annealing to improve parameter search efficiency compared to an exhaustive grid search, and include an intensity-weighted centre of mass of the most optimum points to reduce volatility. We examine two standard classification problems for comparison, and apply the heuristic to bioinformatics and retinal electrophysiology classification.**

I. Introduction

In this article, we address the improvement of generalization performance for highly volatile data sets, such as the classification of unprocessed continuous waveforms generated from retinal electrophysiology which may contain a large proportion of additive noise, or the qualitative analysis of gene and protein sequences based on manual appraisal by bioinformatics experts. We focus on optimal selection of the free parameters in classification using Support Vector Machines (SVM). Specifically, we target optimization of the cost parameter C, which controls the tradeoff between maximization of the margin width and minimizing the number of misclassified samples in the training set [22], and the width γ of the Radial Basis Function (RBF) kernel. Further, we improve generalization performance of the selected model by considering the number of support vectors employed in the model representation, in addition to the prediction accuracy or mean squared error of the model over a test data set.

A. Support Vector Machines

Suppose we are given a set of ℓ observations

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)$$

with inputs $\mathbf{x}_i \in \mathcal{X} = \mathbb{R}^d$, $i = 1, \ldots, \ell$ that indicate targets $y_i \in \mathcal{Y}$. In the general problem of supervised learning, our goal is to find a function $f(\mathbf{x})$ in the set of functions $\mathcal{F}$ which minimizes a loss functional on future observations [4]. For example, we may wish to find a function that minimizes the classification error where $\mathcal{Y} = \{-1, +1\}$, or minimizes the mean squared regression error where $\mathcal{Y} = \mathbb{R}$.

Support Vector Machines [1], [2], [19], [22] map the observations from input space into a higher dimensional feature space using a non-linear transformation, then find a hyperplane in this feature space which optimally separates the known observations by minimizing empirical risk.

In SVM classification, predictions on future observations are then made from [2], [22]

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{\ell} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) - b\right)$$

where b is a numeric offset threshold and $\alpha_i \geq 0$, called the Kühn-Tucker coefficients [1], define the weight of each observation. Those observations where α_i differs from zero are said to be representative *support vectors* [22]. The kernel function $K(\mathbf{x}, \mathbf{x}')$ defines a dot product to transform the observations from input space into feature space. This kernel is chosen *a priori* based on the problem at hand: here we will use the popular RBF kernel

$$K(\mathbf{x}, \mathbf{x}') = e^{-\gamma \left\|\mathbf{x}-\mathbf{x}'\right\|^2}$$

where the free parameter $\gamma > 0$, or *width* parameter [22], controls the width of the Gaussian kernel.

If the classes are separable, the SVM finds the widest possible margin that separates them [1]. However, if the classes have some overlap in their distributions, we may wish to allow some observations to be misidentified in order to find a *soft-margin* hyperplane [22] that optimally separates the remaining samples. This is accomplished by introducing slack variables to the constraints of the Lagrangian optimization problem that forms the SVM training algorithm, effectively imposing an upper bound on the weights of the support vectors such that $0 < \alpha_{sv} \leq C$, where the free parameter $C > 0$ is a *cost* parameter that imposes a penalty on misclassified samples [2].

B. Related Work

Appropriate selection of the free parameters for an SVM is critical for obtaining good performance. Initially, Vapnik [22] recommended direct setting of the kernel parameters and cost function by experts, based on *a priori* knowledge of the particular data set to be evaluated.

Grid searches over an arbitrary range of parameter values is a common technique when such knowledge is unavailable [17], [20], however such searches may be computationally expensive and the precision of the results is subject to the chosen granularity of the grid. In [4], [8], gradient descent methods are proposed based on the minimizing the generalization error, allowing a larger number of parameters to be considered. However, in practical problems such methods

Thomas Trappenberg (corresponding author) and Matthew Boardman are with the Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, B3H 1W5, Canada (phone: 902-494-3087; fax: 902-492-1517; email: tt@cs.dal.ca or Matt.Boardman@dal.ca).

0-7803-9490-9/06/$20.00/©2006 IEEE

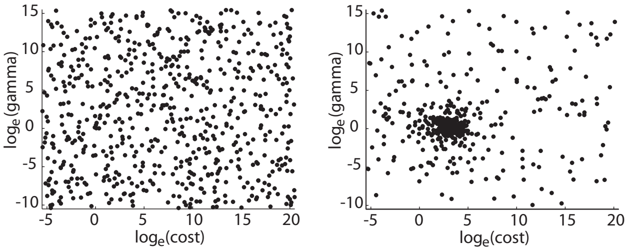

Fig. 1. Comparison of the positions of evaluated points in a uniform random search pattern (*left*) and a stochastic algorithm based on simulated annealing (*right*). Note that although both searches contain exactly 660 evaluations, the simulated annealing approach focuses on the area of interest to a much greater extent. These figures were created from the Bioinformatics data set (Inadequate vs. other) examined later in this paper.

may be affected by the presence of local extrema [12]. This effect may be exacerbated in N-fold cross-validation from random partitioning of the training data. Leave-one-out cross-validation (where $N = \ell$) helps to reduce the effects of local extrema through the complete evaluation of all permutations of the training set at each point in the parameter search [14], but this becomes computationally prohibitive when ℓ is large. In [14], a nearest-neighbor sampling pattern is progressively evaluated as an alternative to gradient descent, but due to the volatile nature of the evaluated surface, the average of multiple locally-optimal models is used, increasing the computational burden.

In [5], an analytical approach is proposed based on rescaling the inputs $\mathbf{x}_i$ in relation to the *span* of the support vectors. For ϵ-insensitive Support Vector Regression (ϵ-SVR) in particular, in [6], [17] analytical approaches to selection of the cost parameter C are proposed based on the mean and standard deviation of the target values y_i. Analytical selection of ϵ is proposed in [22] based on the noise level of the inputs $\mathbf{x}_i$, and in [6] also considering the number of training samples ℓ. In [19], a combination of analytical and combinatorial parameter selection is proposed, such that the choice of ϵ is tuned to a particular noise density but the choice of C is chosen through a numerical approach such as cross-validation.

The well-known parameter optimization method of simulated annealing [13] has recently been proposed as a stochastic method for traversing SVM free parameter space. In Figure 1, a comparison between a purely random search and such a guided, stochastic search is presented. The points evaluated by the simulated annealing algorithm concentrate on the area of interest to a much greater extent. Such techniques have been applied to synthetic and noisy image data for optimization of the cost and kernel parameters [12], feature selection for audio classification with a linear SVM [7] and colon cancer recognition using radial basis function classifiers (RBFC) [23].

While SVM classifiers intrinsically account for a trade-off between model complexity and classification accuracy to enhance generalizability in the non-separable case [22], the generalization performance is still highly dependent on appropriate selection of the C and γ free parameters. Thus, we propose here to take this into account extrinsic to the SVM itself, when tuning these parameters to find the most optimum solution.

II. Visualizing Performance in C, γ Space

If we examine the topology of a surface representing the generalization performance of an SVM classifier, training the classifier using parameters selected by varying the cost parameter C and the width parameter γ of the RBF kernel over a $\log_e$ range of values, some interesting patterns begin to develop. In Figures 2–4, in which light areas correspond to parameter values which yield high accuracy and dark areas those that yield poor accuracy, we see a surface fraught with many sharp local extrema, narrow valleys and sharp cliffs. While the effects of these sharp extrema may well be magnified by the $\log$ operation, we must take them into account since the $\log_e$ space is the surface we wish to traverse. Similar shapes appear for many data sets, for example [11], [20].

The difficulties of traversing such a complex, volatile surface are immediately apparent. Gradient ascent methods [4], [8] may become stuck in local extrema, while hill-climbing algorithms such as the geometric approach in [16] may traverse the space inefficiently. In addition, if we choose an optimum point in parameter space near the edge of a sharp cliff or other local extrema, it is quite possible that small variations in the sample data may cause the surface to subtly shift, causing the classifier to "fall" from the cliff to an area of lower accuracy.

When a high-resolution close-up of a small region of the surface is viewed, smoothed over the mean of several iterations at each point, the chaotic patterns seen at a high level seem to be formed by the convergence of multiple, smaller regions.

III. Proposed Heuristic

In this paper, we use a simulated annealing algorithm [13] to traverse a surface based on the $\log_e$ magnitude of the C, γ free parameters. Here we adapt the continuous, N-dimensional implementation of this algorithm shown in [16], but employ a simple random move generator with a small bias towards the origin $\mathbf{P}_\varnothing$. We also implement occasional restarts to the most optimum points found thus far, with low probability, an alternative suggested in [16].

From this stochastic path through parameter space, adopting the notation of [9], we minimize the cost functional

$$\begin{aligned} \mathscr{E}(f) &= \mathscr{E}_s(f) + \lambda\,\mathscr{E}_c(f) \\ &= \frac{1}{2\ell}\sum_{i=1}^{\ell} |y_i - f(\mathbf{x}_i)| + \lambda\left(\frac{n_{sv}}{\ell}\right)^{\Gamma} \end{aligned} \qquad (1)$$

where the cost $\mathscr{E}(f)$ at the point $\mathbf{P}_i = \{C, \gamma\}$ in parameter space is determined not only by the classification error $\mathscr{E}_s(F) \in [0,1]$ of an SVM classifier trained using the parameters defined by that point, but also by a complexity penalty $\mathscr{E}_c(f) \in (0,1]$ defined by the number of support

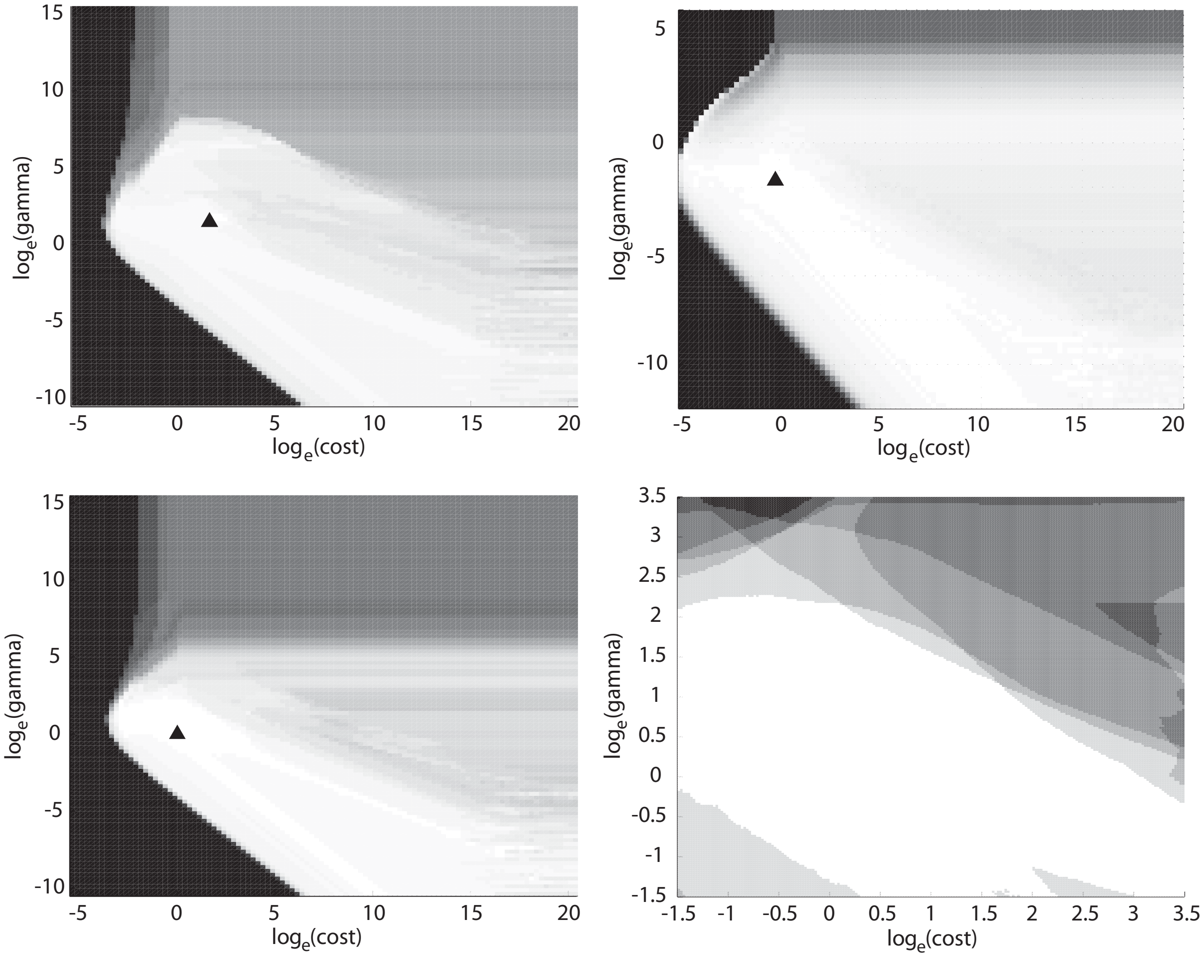

Fig. 2. Error surfaces resulting from SVM classification by varying the C and γ free parameters over a $\log_e$ range of values. Dark areas correspond to high error, whereas light areas correspond to high accuracy. Triangles (▲) indicate the optimum point found through a grid search of $\log_e$ space, considering only the classification accuracy and disregarding model complexity. The *upper right* image shows the result of a typical grid search on the Wisconsin Breast Cancer Database [15]. The remaining images show results from a high-resolution grid search on the three-class Bioinformatics data set examined later in this paper [18]: Valid vs. other (*upper left*), Inadequate vs. other (*lower left*) and a close-up of Inadequate vs. other (*lower right*) showing the convergence of multiple smaller regions. The optimum point is not shown on the close-up for clarity. Some additional examples of this visualization in $\log_e$ space are shown in Figures 3 and 4.

vectors n_{sv} in the model representation of $f(\mathbf{x}_i)$, expressed as a ratio to the total number of observations ℓ. The regularization parameter λ allows control over the tradeoff between classification accuracy and model generalizability. In this paper, we give equal weight to both accuracy and complexity by setting $\lambda = 1$ for each of the following experiments. We also set the free parameter $\Gamma = \frac{1}{2}$ to sharply penalize solutions which obtain high accuracy through high complexity.

Once the cooling schedule has elapsed, we select the absolute best point found $\mathbf{P}_{opt}$. We examine the points surrounding $\mathbf{P}_{opt}$ to select those within a small $\log_e$ radius r_0 and with a cost functional $\mathscr{E} \leq (1+\xi)\mathscr{E}_{opt}$ where $\xi > 0$ is small, then, borrowing a standard method from the field of image processing, we calculate an intensity-weighted centre of mass of these points. This has the effect of reducing the volatility of the resulting end-point arising from the random nature of the generalization surface. The resulting point in parameter space, $\mathbf{P}_{sugg}$, defines the suggested parameters to be used for this particular problem.

Figure 3 (*left*) illustrates the importance of this centre-of-mass operation. Although this can potentially reduce accuracy somewhat in comparison to the optimum point $\mathbf{P}_{opt}$, the resulting point in parameter space is likely to be further from any steep cliffs in the evaluated generalization surface. Since we have a finite set of observations, the decision boundary of the classifier will likely change as additional training samples are evaluated. For example, if we have a large number of observations, it may be prudent to use a small subset of the training data to find optimum parameters, but employ the full set of training data to train the final classifier. This volatility may cause the generalization surface to shift slightly as more samples are added to the training, such that a point in parameter space selected without those samples, close to

TABLE I

SAMPLE RESULTS USING STOCHASTIC HEURISTIC ON WISCONSIN BREAST CANCER AND IRIS PLANT DATABASES [15]. FOR EACH TEST, THE 10-FOLD CROSS-VALIDATION ACCURACY, THE NUMBER OF SUPPORT VECTORS, AND THE NUMBER OF EVALUATED POINTS IN $\log_e$ PARAMETER SPACE ARE SHOWN.

Database	Search Method	Acc.	n_{sv}	Evals.
WBCD	Fast-Cooling Heuristic	96.5	36	660
	Slow-Cooling Heuristic	96.0	34	6880
	Grid Search: Best	97.4	129	7373
	Grid Search: Suggested	97.1	77	7373
Iris database:	Fast-Cooling Heuristic	100	3	660
Iris setosa	Slow-Cooling Heuristic	100	3	6880
(linear)	Grid Search: Best	100	12	7373
	Grid Search: Suggested	100	12	7373
Iris	Fast-Cooling Heuristic	95.3	13	660
Iris versicolour	Slow-Cooling Heuristic	94.7	9	6880
(non-linear)	Grid Search: Best	98.0	28	7373
	Grid Search: Suggested	96.7	35	7373
Iris database:	Fast-Cooling Heuristic	96.7	8	660
Iris virginica	Slow-Cooling Heuristic	98.0	6	6880
(non-linear)	Grid Search: Best	98.0	33	7373
	Grid Search: Suggested	97.3	35	7373

an edge such as illustrated here, may "fall" from the edge to a region of lower accuracy.

Details of the specific implementation of the heuristic used in this paper are summarized in Appendix I.

IV. RESULTS

We have obtained reasonable results with this heuristic by setting the arbitrary cooling schedule to start at $T_0 = 100$ and cool to $T_C = 0.1$, reducing the temperature every $m = 10$ iterations by $\delta = 0.01$ (for *slow* cooling) or by $\delta = 0.1$ (for *fast* cooling). The fast cooling schedule therefore results in 660 evaluations, whereas the slow cooling schedule results in 6880 evaluations. We chose these schedules such that the number of evaluations would approximately match those of coarse- and fine-grained grid searches over the same parameter space, and did not tune these schedules for any particular experiment as we desire an automatic method for parameter selection.

In each of the following experiments, the origin bias was set to $\alpha = 0.1$ (one tenth the magnitude of the move selected by random walk). The probability of accepting a test point with a detrimental cost was set to $p_{acc} = 0.1$, so long as the cost is within $\beta = 0.1$ of the current point. At any point in the search, the probability of abandoning the current path in favour of a random selection amongst the best points found so far was $p_{res} = 0.01$. The intensity-weighted centre of mass calculation after the completion of the cooling schedule included those points within a $\log_e$ radius of $r_0 = 1$ from the best point found $\mathbf{P}_{opt}$, and which have a cost functional $\mathscr{E} \leq (1 + \xi)\mathscr{E}_{opt}$ where $\xi = 0.02$. No fine-tuning of these parameters was performed for any particular experiment, since we wish to evaluate generalization performance when the heuristic is employed blindly.

Empirically, we have found that the noisy nature of N-fold cross-validation has made little appreciable difference in the results when using the stochastic approach, since points near the end of the search may be quite nearby as the temperature T decreases. However, one often-overlooked step which can have a significant effect when using SVM is to ensure the independently and identically distributed (i.i.d.) inputs necessary for optimal classification [2], [22]. For example, one can centre and scale the inputs such that all dimensions have zero mean and values $x_i \in [-1, +1]$. However, when faced with noisy, volatile input data such as the sensor waveforms examined in Section IV-D, there may be peaks in future observations that were not seen in the training data, thereby breaking these arbitrary bounds on the input vector. A common alternative is to centre and scale to zero mean and unit variance, but in our tests we found this reduced the accuracy somewhat.

The following experiments were conducted in MATLAB (Mathworks) using LIBSVM [3] version 2.81.

TABLE II

CONSISTENCY OF RESULTS FOR IRIS DATABASE, *Iris setosa* CLASS (LINEARLY SEPARABLE). THE RESULTS SHOWN FOR THE FAST- AND SLOW-COOLING HEURISTICS ARE THE MEAN OVER TEN RUNS.

Search Method	$\log_e(\gamma)$	$\log_e(C)$	Accuracy	n_{sv}
Fast-Cooling Heuristic	-1.83	2.80	100	3
(Standard Deviation)	(0.05)	(0.14)	(0)	(0)
Slow-Cooling Heuristic	-1.81	2.76	100	3
(Standard Deviation)	(0.02)	(0.02)	(0)	(0)
Grid Search: Best	0.00	0.00	100	12
Grid Search: Suggested	-0.15	0.06	100	12

TABLE III

CONSISTENCY OF RESULTS FOR IRIS DATABASE, *Iris virginica* CLASS (NON-SEPARABLE). THE RESULTS SHOWN FOR THE FAST- AND SLOW-COOLING HEURISTICS ARE THE MEAN OVER TEN RUNS.

Search Method	$\log_e(\gamma)$	$\log_e(C)$	Accuracy	n_{sv}
Fast-Cooling Heuristic	-6.8	15.6	95.8	8.3
(Standard Deviation)	(2.02)	(3.62)	(0.89)	(0.5)
Slow-Cooling Heuristic	-8.5	18.4	96.4	7.7
(Standard Deviation)	(0.66)	(0.56)	(1.05)	(1.2)
Grid Search: Best	-1.0	0.8	98.0	33
Grid Search: Suggested	-1.9	1.3	97.3	35

A. Classic Classification Problems

We first apply this heuristic to two standard classification problems, in order to compare the results of the stochastic heuristic above including the model complexity measure, with a reasonably-sized grid search based only on cross-validation classification accuracy.

The Wisconsin Breast Cancer Database (WBCD) [15] is a binary classification problem with non-separable data, containing 699 instances and nine discrete attributes. Instances with missing data were removed, leaving 683 observations. We left the classes unbalanced, with the natural class distribution, but centered and scaled all numeric attributes based on the mean and maximum magnitude of each attribute, such that $\mathbf{x}_i \in [-1, +1]$ in order to approximate i.i.d. data.

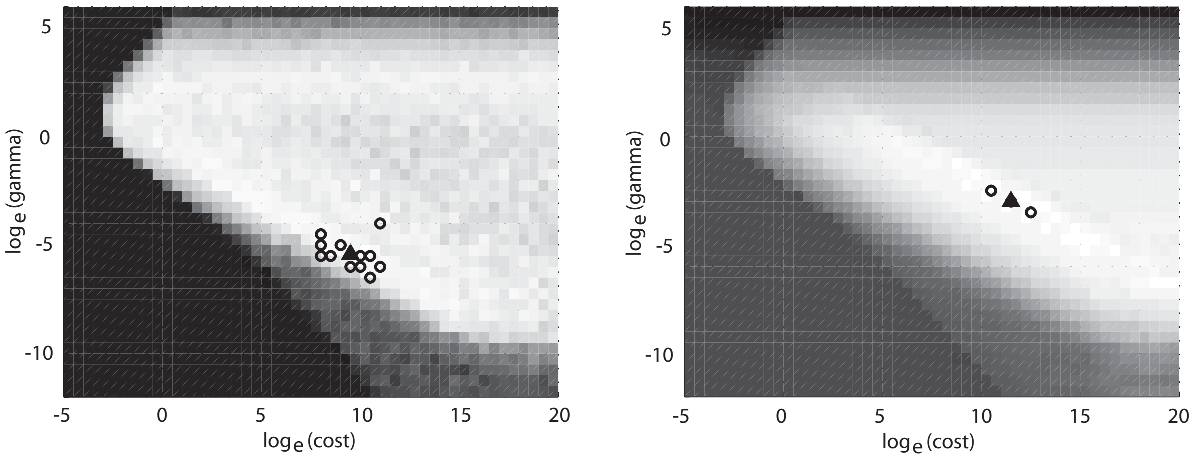

Fig. 3. Visualizing the generalization performance of a grid search through $\log_e$ space for the *Iris versicolour* class of the Iris Plant Database [15]. Circles (○) indicate the group of best points found. The triangle (▲) shows the optimum point calculated from this group, using an intensity-weighted centre-of-mass operation. On the *left*, the generalization performance is plotted as in Fig. 2 and 4, however on the *right*, a complexity penalty is added from Equation 1. The resulting surface is smoother, indicating that the number of support vectors in the complexity penalty is less volatile than the cross-validation accuracy. On the *left*, we see an example of how the centre-of-mass operation moves the optimum point further away from a region of lower accuracy, improving generalization performance.

The Iris Plant Database [15] includes 50 instances for each of three classes, for a total of 150 observations, with four continuous-valued numeric attributes. The *Iris setosa* class is known to be linearly separable from the others, *Iris versicolour* and *Iris virginica* [15]. In our tests, we again left the classes unbalanced with the natural distribution, and centered and scaled all attributes as above.

Sample results from these tests are summarized in Table I and compared to a reasonably-sized grid search as shown in Figures 2 (*upper right*), 3 (*right*) and 4. We find that the heuristic results in a model with comparable accuracy, often obtained with fewer calculations. Due to our inclusion of the number of support vectors n_{sv} when evaluating the cost functional at each point, the resulting models all have lower complexity than that obtained through a grid search using cross-validation accuracy as the only measure. For example, the slow-cooling heuristic for the *Iris virginica* classifier achieved 98.0% 10-fold cross-validation accuracy with six support vectors, whereas the best point from a grid search yields the same accuracy with 33 support vectors.

Some sacrifice of accuracy may be necessary as a tradeoff to favour low complexity: for example, the WBCD classifier with the slow-cooling heuristic obtained 96.0% 10-fold cross-validation accuracy, whereas the grid search obtained 97.4%. On closer examination, however, we see that this slightly higher accuracy was obtained at the expense of high complexity: the grid search results required 129 support vectors, whereas the slow-cooling heuristic required only 34, representing a significant reduction in model complexity.

B. Consistency of Results

Since the heuristic takes a random path through parameter space, we may wish to determine how consistent the results are when run several times on the same data set.

For this purpose, we use the *Iris setosa* and *Iris virginica* classes from the Iris data set, which are linearly separable and non-separable respectively. The results from these tests are shown in Figure 4 and summarized in Tables II, III. Notice that both the slow- and fast-cooling heuristics obtain nearly the same accuracy as the grid search, but with far fewer support vectors, indicating that the resulting model is much less complex. The results are reasonably consistent with the slower cooling rate, but allow for higher variability with the faster cooling rate.

Although the linear separability of the *Iris setosa* class allows a wide range of values which will achieve 100% accuracy with a very low number of support vectors, the suggested parameters for both the fast- and slow-cooling heuristics overlap. For the non-separable problem, however, the variability is relatively high for the fast-cooling heuristic. The range of suggested parameter values for the slow-cooling heuristic is much more narrow, indicating that for non-separable data, a slower cooling schedule should be used.

C. Bioinformatics

The Bioinformatics data set [18] includes three measures (gap ratio g, normalized site log likelihood ratio h and consistency index CI) used to determine the quality of alignment of 17 821 gene and protein sequences in preparation for phylogenetic analysis (12 625 of these were available to us for this experiment). A fourth measure, the site rate, was available in our test set but was not used in our experiments, so that we could compare our results with [18]. The alignment quality of each sequence was categorized by experts into three classes: *Valid*, *Inadequate* and *Ambiguous*. The authors of [18] compared the classification performance of SVM with a C4.5 decision tree algorithm (C4.5) and a Naïve Bayesian classifier (NB), and found inferior performance for SVM.

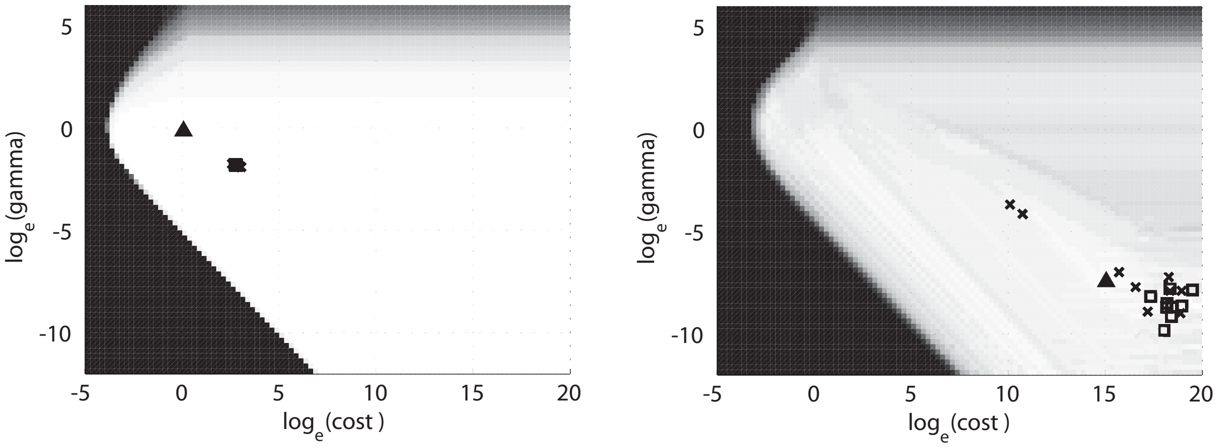

Fig. 4. Consistency of results from ten sample runs for the Iris database, comparing the results from the slow- and fast-cooling heuristic with a grid search: the linearly-separable *Iris setosa* class (*left*) and the non-separable *Iris virginica* class (*right*). The grid search evaluated 7373 points in parameter space, whereas the fast-cooling heuristic evaluated 660 and the slow-cooling heuristic evaluated 6880. The grid search considered only cross-validation accuracy, whereas the heuristics considered both cross-validation accuracy and model complexity. Solid triangles (▲) indicate the suggested optimum point resulting from the grid search: the accuracy-weighted centre of mass of the point with best overall cross-validation accuracy that is closest to the origin $\mathbf{P}_\varnothing$. Squares (□) indicate suggested positions from the slow-cooling heuristic, while crosses (×) indicate suggested positions from the fast-cooling heuristic. In the highly separable case (*left*), both the fast- and slow-cooling heuristics find the same optimum points, to the lower right of the grid search which does not consider model complexity. However, in the non-separable case, the heuristics have more difficulty converging to a point, as can be seen by the higher spread in the distribution of optimum points found. A slower cooling schedule might correct for this, as the points found by the slow-cooling schedule are much more closely spaced than those of the fast-cooling schedule.

However, they mentioned that their SVM implementation did not attempt to identify optimal parameters.

In our experiments with this data set, we used $(1 - \frac{1}{CI})$ rather than CI in order to more closely match the distribution of the other dimensions, which are heavily zero-weighted. Since all values in the set are positive, all three dimensions were then scaled such that $\mathbf{x}_i \in [0, 1]$, but were not centred on their means. Three binary SVM classifiers were trained independently on randomly-selected but class-balanced subsets of 100 sequences — 50 of the class to be selected and 50 of any other class with the natural class distribution — using the above heuristic to determine optimal γ and C parameters for each classifier.

The results of these tests are summarized in Table IV. We found the cross-validation accuracy of the Inadequate vs. other classifier to be comparable with that reported in [18]. However, with proper parameter selection through either grid search or through our heuristic, we found that the results for the Valid class were favourable to the results for both the SVM with default parameters and the probabilistic classifiers reported in [18].

Due to the high accuracy of the Inadequate vs. other and Valid vs. other classifiers, we decided to train a fourth classifier to distinguish Inadequate vs. Valid, ignoring the Ambiguous class during training. The results from these tests are also shown in Table IV. The classifier was trained with 100 randomly-selected (but class-balanced) points, as with the other classifiers. This resulted in high accuracy and consistency over ten runs of the fast-cooling heuristic with a different, randomly-selected set of training samples for each run. The fast-cooling heuristic was then trained on 1000 similarly-selected points, and achieved a 10-fold cross-validation accuracy of 99.5% with only 10 support vectors. Further investigation may discover whether using this classifier to predict the validity of the ambiguous alignments may outperform manual assignment.

D. Retinal Electrophysiology

Waveform classification is a challenging problem as the number of dimensions may be much larger than the number of available observations ($\ell \ll d$): with such a large number of dimensions, it is statistically more likely that one or more of those dimensions may be fully-separable simply by chance, and therefore any classifier may base predictions on only that dimension during training. This is known as the "the curse of dimensionality," and is a significant advantage for SVM [19]. To reduce the effects of this problem, proper cross-validation is critical for generalization performance: in this classification experiment, due to the relatively low number of samples, leave-one-out cross-validation was used instead of N-fold cross-validation, resulting in a smoother error surface.

The retinal electrophysiology data set used in this paper comes from pattern electroretinography (PERG). Axotomy procedures were performed on female domestic pigs (*Sus domesticus*), each approximately six months of age, as part of an ongoing medical research project examining the electrophysiological contributions of bipolar and ganglion cells of the retina. A control PERG measurement under ketamine anaesthesia was first taken from each subject for comparison. The axotomy procedure then severed the optic nerve. The time between the axotomy procedure and PERG

TABLE IV

10-FOLD CROSS-VALIDATION ACCURACY RESULTS FROM A GRID SEARCH AND FROM THE STOCHASTIC HEURISTIC FOR THE BIOINFORMATICS DATA SET WITH BALANCED TRAINING SAMPLES, COMPARED TO THE RESULTS OBTAINED IN [18]. NaN INDICATES ALL VALUES WERE CLASSIFIED AS OTHER. RESULTS FOR VALID VS. INADEQUATE (100 SAMPLES) ARE THE MEAN OVER TEN RUNS.

Class	Search Method	Acc.
Valid vs. other	Heuristic Grid Search SVM in [18] C4.5 in [18] NB in [18]	91.0 95.0 NaN 87.2 55.4
Inadequate vs. other	Heuristic Grid Search SVM in [18] C4.5 in [18] NB in [18]	98.0 98.0 97.0 93.8 96.6
Valid vs. Inadequate: 100 samples	Heuristic Mean (Standard Deviation)	99.1 (0.9)
Valid vs. Inadequate: 1000 samples	Sample run	99.5

TABLE V

SAMPLE RESULTS FROM RETINAL ELECTROPHYSIOLOGY CLASSIFICATION, COMPARING A GRID SEARCH TO THE SLOW-COOLING HEURISTIC. THE MEAN OVER 28 COMBINATIONS OF 12 BALANCED SUBSETS OF 14 SUBJECTS ARE SHOWN.

Search Method	Acc.	n_{sv}	Evals.
Fast-Cooling Heuristic	98.5	6.2	660
Slow-Cooling Heuristic	98.5	6.2	6880
Grid Search	99.4	8.5	6603

measurements was approximately six weeks, in order to allow phagocytosis processes to completely consume any remaining ganglion cells in the retina. Gain settings were held constant across all measurements. Data from 103 high-contrast chequer locations displayed on a 75 Hz source were averaged using an m-sequence over approximately 2.5 minutes. This procedure results in a $103 \times 145 = 14\,935$ point vector for each observation. A mean of all chequer locations was generated to form a 145 point waveform, corresponding to 145 ms at a 1000 Hz sampling rate. A discussion of similar methods may be found in [21].

The preliminary data used in this paper has 14 observations in two classes: six *axotomy* and eight *control*. The mean waveforms for each class are shown in Figure 5. To perform the classification, the raw waveform was input as a 145-dimensional vector. Each dimension was centred and scaled independently by the mean and magnitude, such that the resulting inputs had a mean of zero and values $\mathbf{x}_i \in [-1, +1]$. To more closely approximate a balanced data set, the classifications were performed on each of the possible $\binom{8}{6} = 28$ combinations of 12 balanced subsets, and the resulting accuracy was taken as the mean across all 28 runs.

The results of the classification are shown in Table V. We have found excellent generalization performance using the heuristic with this data set: the fast-cooling heuristic found the same accuracy and complexity as the slow-cooling heuristic but with far fewer evaluations.

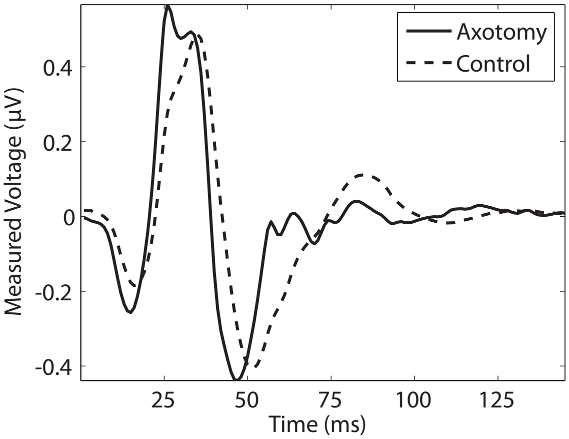

Fig. 5. Mean pattern ERG waveforms for six axotomy and eight control subjects. The mean waveforms appear to be visually separable, but even to an expert, the actual observed waveforms may be somewhat ambiguous upon visual examination.

V. CONCLUSIONS

Support vector machines are robust, but naïve choices of the free parameters will often result in unacceptable generalization error. Appropriate selection of free parameters is essential to achieving high performance.

In this paper, we have kept the simulated annealing scheme quite simple, with a choice of two cooling schedules: *fast* cooling ($\delta = 0.1$) and *slow* cooling ($\delta = 0.01$). The slow cooling schedule allows a more extensive search of the parameter space and may be useful for finding global, narrow extrema, whereas the fast cooling schedule may be used to find a good solution quickly. Using the proposed stochastic heuristic to navigate the error surface from two classical classification problems and two real classification problems, with selection based on the inclusion of a model complexity measure and an intensity-weighted centre of mass, we have found results comparable with those of a grid-search but with lower model complexity. The technique can easily be extended to take further free parameters into account, for example to discover an optimal solution in a three-dimensional parameter space defined by C, γ and the width of the ϵ-tube used in the soft-margin loss function for regression or function estimation problems [19], [22].

We find that when tuning free parameters, including a model complexity penalty enhances the generalizability of the final solution. Blind application of this simple annealing scheme was found to give good results with several classic and real data sets. Further analysis is warranted to determine the generality of this approach with a wider array of practical problems, and to compare the results of this heuristic with other parameter optimization methods, such as [4], [14], [20].

APPENDIX I

The details of the simulated annealing heuristic used in this paper are as follows. A MATLAB (Mathworks) implementation may be downloaded from the authors' web site (`http://www.cs.dal.ca/~tt`).

1. Initialize a high temperature $T \leftarrow T_0$, and set the starting point $\mathbf{P}_i \leftarrow \mathbf{P}_0$ within the $\log_e$ parameter space.

2. Determine a new test point $\mathbf{P}_t$, taken in a random direction from $\mathbf{P}_i$ (with uniform distribution), and a random scalar distance (with normal distribution) multiplied by the ratio T/T_0 and the width (or height) of the parameter space.

3. Add a small origin bias $\mathbf{P}_t \leftarrow \alpha(\mathbf{P}_i - \mathbf{P}_\varnothing)$ where $\mathbf{P}_\varnothing$ defines the $\log_e$ origin and α is a small scalar. Check that the current point $\mathbf{P}_t$ lies within the boundary conditions of the parameter space: if not, select a new point at random anywhere within the parameter space.

4. Determine a scalar cost functional $\mathscr{E}_t$ for the current position $\mathbf{P}_t$, including the classification error and model complexity of an SVM model trained using the parameters at this point. If the cost functional $\mathscr{E}_t$ for $\mathbf{P}_t$ is less than that of $\mathscr{E}_i$ of $\mathbf{P}_i$ (or if this is the first evaluated point), accept this point as the new position in the parameter space. If not, but the resulting ascent in cost is small, perhaps accept this point with a small probability p_{acc}. Otherwise, reject it.

5. If the point was accepted, set $\mathbf{P}_i \leftarrow \mathbf{P}_t$ and $\mathscr{E}_i \leftarrow \mathscr{E}_t$, then compare the cost functional $\mathscr{E}_i$ with $\mathscr{E}_{opt}$ of the most optimum points obtained thus far: if the cost is lower, replace the existing list with the current point; if the cost is approximately equivalent within a small margin of error, add the current point to the existing list.

6. If the cost $\mathscr{E}_i > (1+\beta)\mathscr{E}_{opt}$, where β is a small scalar, then with a very small reset probability $p_{res} \ll p_{acc}$, jump to a randomly selected point from the list of optimum points.

7. Drop the temperature $T \leftarrow T(1-\delta)$ every m iterations. If the temperature is still higher than the termination criteria T_C, continue through further iterations from step 2. Otherwise, determine a single optimum point $\mathbf{P}_{opt}$ as the point in the set of optimum points that lies closest to the origin $\mathbf{P}_\varnothing$.

8. Gather a set of points from the list of all evaluated points which have a cost within $\mathscr{E} \leq (1+\xi)\mathscr{E}_{opt}$ of the best cost $\mathscr{E}_{opt}$, where ξ is a small scalar, and which lie within a small radius r_0 from the optimum point $\mathbf{P}_{opt}$.

9. Determine the suggested point $\mathbf{P}_{sugg}$ as the intensity-weighted centre of mass of this set of points, using $(1-\mathscr{E}_i)$ as the intensity for each point i, and retrain the model with the parameters determined by this suggested point using all training points.

ACKNOWLEDGMENT

The authors would like to thank Dr. François Tremblay, Dr. Denis Falvey and Dr. Christian Blouin, Dalhousie University, for access to the data used in this paper, and the anonymous reviewers for many helpful suggestions. This work was supported in part by the NSERC grant RGPIN 249885-03.

REFERENCES

[1] B. E. Boser, I. M. Guyon and V. N. Vapnik, "A training algorithm for optimal margin classifiers," *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pp. 144–152, 1992.

[2] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[3] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 2001.

[4] O. Chapelle, V. Vapnik, O. Bousquet and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1–3, pp. 131–159, 2002.

[5] O. Chapelle and V. Vapnik, "Model selection for support vector machines," in *Advances in Neural Information Processing Systems*, Cambridge, MA: MIT Press, 1999.

[6] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Networks*, vol. 17, no. 1, pp. 113–226, 2004.

[7] S. Degroeve, K. Tanghe, B. De Baets, M. Leman and J.-P. Martens, "A simulated annealing optimization of audio features for drum classification," In *Proceedings of the 6th International Conference on Music Information Retrieval*, London, UK, September 2005.

[8] R.-E. Fan, P.-H. Chen and C.-J. Lin, "Working set selection using the second order information for training SVM," *Journal of Machine Learning Research*, vol. 6, pp. 1889–1918, 2005.

[9] S. Haykin, *Neural Networks: A Comprehensive Foundation*, New York: Macmillan, pp. 245–255, 1994.

[10] D. Heckerman, "A tutorial on learning with Bayesian networks," In *Learning in graphical models*, Cambridge, MA: MIT Press, pp. 301–354, 1999.

[11] C.-W. Hsu, C.-C. Chang and C.-J. Lin, "A practical guide to support vector classification," Technical report available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 2003.

[12] F. Imbault and K. Lebart, "A stochastic optimization approach for parameter tuning of support vector machines," In *Proceedings of the of the 17th International Conference on Pattern Recognition (IPCR)*, vol. 4, pp. 597–600, 2004.

[13] S. Kirkpatrick, C. D. Gelatt, Jr. and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[14] M. Momma and K. P. Bennett, "A pattern search method for model selection of support vector regression," In *Proceedings of the 2nd SIAM International Conference on Data Mining*, Philadelphia, Pennsylvania, 2002.

[15] D. J. Newman, S. Hettich, C. L. Blake, C. J. Merz, *UCI Repository of machine learning databases*, Available at `http://www.ics.uci.edu/~mlearn`, Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing, 2nd ed.*, Cambridge University Press, pp. 444–455, 1992.

[17] B. Schölkopf, C. J. C. Burges and A. Smola, *Advances in Kernel Methods: Support Vector Learning*, Cambridge, MA: MIT Press, 1999.

[18] Y. Shan, E. E. Milios, A. J. Roger, C. Blouin and E. Susko, "Automatic recognition of regions of intrinsically poor multiple alignment using machine learning," *Proceedings of the 2003 IEEE Computational Systems Bioinformatics Conference*, pp. 482–483, August 2003.

[19] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[20] C. Staelin, "Parameter selection for support vector machines," Technical Report HPL-2002-354 (R.1), HP Laboratories Israel, 2003.

[21] E. Sutter and D. Tran, "The field topography of ERG components in man—I. The photopic luminance response," *Vision Research*, vol. 32, no. 3, pp. 433–446, 1992.

[22] V. N. Vapnik, *The Nature of Statistical Learning Theory, 2nd ed.*, New York: Springer-Verlag, 1999.

[23] H.-Q. Wang, D.-S. Huang and B. Wang, "Optimisation of radial basis function classifiers using simulated annealing algorithm for cancer classification," *Electronics Letters*, vol. 41, no. 11, 2005.